

Using XML / XSLT to Format Generic PDF Reports from Variable HTML Table data

Proof of Concept

Rich Casella

rac@bnl.gov

Brookhaven National Laboratory

ITD Application Services

BROOKHAVEN
NATIONAL LABORATORY

a passion for discovery



U.S. DEPARTMENT OF
ENERGY

Office of
Science

Background

- 30 Years at Brookhaven
- Programmer
 - Degrees in CS and Applied Mathematics
 - Accelerator Control Systems
 - System Administration
 - Application Services
- System Administration (Unix/Linux)
 - Scientific Environment
 - Configuration Management
 - Enterprise IT Infrastructure
- Cyber Security
 - DOE Audits
 - Data Calls
- Application Services
 - Collaborative Tools/Packages
 - Stand-alone Applications

The Problem

Reports Needed for Variable Data

- Increased need for reports from configuration data
- Configuration Management reports vary widely depending on the audience and data requested
 - DOE Auditors
 - Cyber Security
 - System Administrators
- Usually formatted into tables with fixed columns, but the number of columns can vary
- Types of data collected (many possible fields from each type)

NIC	OS	Packages	Accounts
Contact	Host Info	Processes	Files
Config	Scan	Department	

The Problem

...the possession of great power necessarily implies great responsibility

- Thomas Curson Hansard (not Stan Lee)

I need a report showing all Unix/Linux machines on site that are running an unsupported version of openssh, including machine name, what version it is running, and contact information, sorted by department code by COB today.

- Put together query
- Get data
- Configure/send report
- Repeat

Need to simplify!

Vision

Eliminate the middle man (me)

- Programmers (me) use tools to provide CGI interfaces to the data
- Consumers manipulate data till they are satisfied
- Report sent at the push of a button

Need more *tools* to make this happen

Existing Tools to collect/display data

Perl code snippet (simplified just a bit)

Comments in blue, orange text indicates library modules called.

The content we are interested in is marked in red.

```
# Get all known client data
&cache::getClientData(\%clients, $ndays, $dept);

# Get a database connection
my $dbh = &dblib::connectDBNew("$dbInstance", "$userName");
# and the data from that connection that we are interested in
&genericDB::getGenericRSETData($dbh, 'E$GET_OUT_OF_DATE_SCANS', \@rtnData);

# Local function to correlate and format the data,
# saved into $content in POSH HTML TABLE format
&formatData(\%clients, \@rtnData, \<strong>$content</strong>);

# Display the web page using the templates and BNL style sheets
# Content can be anything, including HTML,
# It is displayed as the contents of a <TD> tag
&wwwTemplates::displayPage("<strong>$content</strong>", \%params, \@vertMenu);
```

Resulting Web Page (small portion)

The screenshot shows a web browser window with the title "Ordo Client Check". The address bar contains the URL "http://ordo.bnl.gov/cgi/res/dbUpdates.pl". The browser interface includes navigation buttons (Back, Forward, Reload, Stop), a search box, and a print button. The page header features the "Information Technology Division" logo and the "BROOKHAVEN NATIONAL LABORATORY" logo with a "Home" link. A sidebar on the left contains a search box and a menu with items: "BNL Home", "ITD Home", "Other Information" (highlighted), "BNL Site Index", and "Can't View PDFs?".

Non Compliant Ordo Clients

The Ordo clients listed on this page are non-compliant one way or another regarding the frequency of updates that they are reporting to the database. Note that only machines that have been seen on the network in the past 2 weeks are listed here.

Definitions of non-compliance are defined as follows:

1. Client has never successfully sent a fast, full, or heartbeat
2. Client has never done a software update
3. A fast scan or heartbeat has not been received from the client within the past hour
4. A heartbeat has not been received within the past hour
5. A full scan has not been received within the past 2 weeks
6. The latest client software is not running on the client machine

Major non-compliance issues are listed in red.
Minor non-compliance issues are listed in amber.

GPG ID	Last Fast Scan	Last Full Scan	Last Heartbeat	Last Update	Version
B2F926DE	04-OCT-2009	19-OCT-2009	20-OCT-2009	Up To Date	Up To Date
53BDE375	17-MAR-2009	20-OCT-2009	20-OCT-2009	Up To Date	Up To Date
305C2B8A	09-MAR-2009	19-OCT-2009	20-OCT-2009	Up To Date	Up To Date
A25C1966	17-MAR-2009	20-OCT-2009	20-OCT-2009	Up To Date	Up To Date
DB6717E3	29-SEP-2009	19-OCT-2009	20-OCT-2009	Up To Date	Up To Date

Possible Report Formats

- Plain text: No
- Delimited text: Good for scripting, not reports
- XL Spreadsheet: Too much manual work, MS centric (not that that's a bad thing)
- HTML: a convenient (easy) way to display report data, but not an optimal medium for distribution
- PDF: Ok, but how do we create it on the fly?

HTML => XML => PDF via XSLT

XSL Transformation (XSLT) is a declarative XML based language used for the transformation of XML into new documents, usually for some sort of publication.

Common Formats

- XML/(X)HTML
- Plain Text
- PDF
- WML
- Postscript

Other possible uses for XSLT!

What do we need?

- Simplify the problem
 - Restrict the problem initially to the report only
 - Restrict the transformation to well-formed HTML Table
 - Can always be expanded later if need is determined
- XML representation of the data
 - Simple Perl routine to transform the HTML table into XML
- XML Schema Definition File (XSD)
- XSLT to define the XML transformation
- Server tools to perform transformation
 - Xalan - XSLT processor
 - FOP - Formatting Object Processor

Platform/Software Requirements

Mine, not necessarily yours

- Unix/Linux FreeBSD derivative
- Apache Web Server
- Perl Common Gateway Interface (CGI Forms)
- Xalan: Apache XSLT processor for transforming XML documents into other document formats
- FOP: Apache Formatting Objects Processor is a print formatter that renders resultant XML documents into various output formats
- XML/XSLT editing tool

XML Editor

Some things are just better done with COTS tools

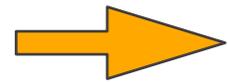
- Oxygen: <http://www.oxygenxml.com/>
- Not particularly expensive (~ \$350.00)
- Not the only one out there
- Full featured editor
 - SVN support
 - XSD/XSD/XSLT, transformations (Xalan, Saxon), code generation, documentation generation, tag completion, full featured debugger, etc.
- Maybe the best, maybe not, it's the one I use

XML Schema (XSD) Development

```
<xs:element name="XMLReport">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ReportHeader" type="ReportHeaderType"/>
      <xs:element name="ReportTable" type="Table"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:complexType name="ReportHeaderType">
  <xs:all>
    <xs:element name="ReportTitle" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Author" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="DeptCode" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="date" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Source" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Summary" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:all>
</xs:complexType>
```

XML Schema (XSD) Development



```
<xs:element name="XMLReport">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="ReportHeader" type="ReportHeaderType"/>  
      <xs:element name="ReportTable" type="Table"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

```
<xs:complexType name="ReportHeaderType">  
  <xs:all>  
    <xs:element name="ReportTitle" type="xs:string" minOccurs="0" maxOccurs="1"/>  
    <xs:element name="Author" type="xs:string" minOccurs="0" maxOccurs="1"/>  
    <xs:element name="DeptCode" type="xs:string" minOccurs="0" maxOccurs="1"/>  
    <xs:element name="date" type="xs:string" minOccurs="0" maxOccurs="1"/>  
    <xs:element name="Source" type="xs:string" minOccurs="0" maxOccurs="1"/>  
    <xs:element name="Summary" type="xs:string" minOccurs="0" maxOccurs="1"/>  
  </xs:all>  
</xs:complexType>
```

XML Schema (XSD) Development

```
<xs:complexType name="Table">  
  <xs:sequence>  
    <xs:element name="TCaption" type="xs:string" minOccurs="0"/>  
    <xs:element name="TNumCols" type="xs:int"/>  
    <xs:element name="Orientation" type="xs:string"/>  
    <xs:element name="TCols" type="TColHeaders"/>  
    <xs:element name="TRow" type="TRow" maxOccurs="unbounded"/>  
    <xs:element name="TFooter" type="xs:string" minOccurs="0"/>  
  </xs:sequence>  
</xs:complexType>
```

```
<xs:complexType name="TColHeaders">  
  <xs:sequence>  
    <xs:element name="TColHeader" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>  
  </xs:sequence>  
</xs:complexType>
```

```
<xs:complexType name="TRow">  
  <xs:sequence>  
    <xs:element name="TData" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>  
  </xs:sequence>  
</xs:complexType>
```

XML Schema (XSD) Development



```
<xs:complexType name="Table">  
  <xs:sequence>  
    <xs:element name="TCaption" type="xs:string" minOccurs="0"/>  
    <xs:element name="TNumCols" type="xs:int"/>  
    <xs:element name="Orientation" type="xs:string"/>  
    <xs:element name="TCols" type="TColHeaders"/>  
    <xs:element name="TRow" type="TRow" maxOccurs="unbounded"/>  
    <xs:element name="TFooter" type="xs:string" minOccurs="0"/>  
  </xs:sequence>  
</xs:complexType>
```

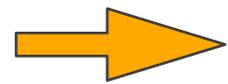
```
<xs:complexType name="TColHeaders">  
  <xs:sequence>  
    <xs:element name="TColHeader" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>  
  </xs:sequence>  
</xs:complexType>
```

```
<xs:complexType name="TRow">  
  <xs:sequence>  
    <xs:element name="TData" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>  
  </xs:sequence>  
</xs:complexType>
```

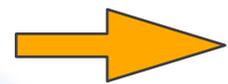
XML Schema (XSD) Development

```
<xs:complexType name="Table">  
  <xs:sequence>  
    <xs:element name="TCaption" type="xs:string" minOccurs="0"/>  
    <xs:element name="TNumCols" type="xs:int"/>  
    <xs:element name="Orientation" type="xs:string"/>  
    <xs:element name="TCols" type="TColHeaders"/>  
    <xs:element name="TRow" type="TRow" maxOccurs="unbounded"/>  
    <xs:element name="TFooter" type="xs:string" minOccurs="0"/>  
  </xs:sequence>  
</xs:complexType>
```

```
<xs:complexType name="TColHeaders">  
  <xs:sequence>  
    <xs:element name="TColHeader" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>  
  </xs:sequence>  
</xs:complexType>
```



```
<xs:complexType name="TRow">  
  <xs:sequence>  
    <xs:element name="TData" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>  
  </xs:sequence>  
</xs:complexType>
```



New XML Library Calls

Two Step Process

```
# Transform the content to XML, adding header information for report
```

```
&xml::HTMLTable2XML(  
    "XML Transformation Test",      # Report Title  
    "Rich Casella",                # Author  
    "AO",                           # Department Code  
    "$date",                        # Date  
    "Ordo Database",               # Content Source  
    "$summary",                    # Report Summary  
    "XML TEST",                    # Table Caption  
    1,                              # 1=Portrait, 0=Landscape  
    6,                              # Number of Columns  
    $content,                       # Data for report  
    \$xml                           # XML transformed from HTML  
);
```

```
# Transform the XML to PDF and email it to recipient(s)
```

```
&xml::XMLTable2PDF($xml, 'rac@bnl.gov');
```

XML Development

Header

```
<?xml version="1.0" encoding="UTF-8" ?>
<XMLReport xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file:/Applications/oxygen/Report.xsd">
  <ReportHeader>
    <ReportTitle>XML Transformation Test</ReportTitle>
    <Author>Rich Casella</Author>
    <DeptCode>AO</DeptCode>
    <date>August 2, 1975</date>
    <Source>Ordo Database</Source>
    <Summary>This is a test of HTML to PDF translation using XML as the mark up language and
      XSLT and the transformation mechanism. All HTML code is transformed to a standard format,
      saved to a temporary file, and emailed to the chosen recipient.</Summary>
  </ReportHeader>
```

XML Development Table

```
<ReportTable>
  <TCaption>XML TEST</TCaption>
  <TNumCols>6</TNumCols>
  <Orientation>Portrait</Orientation>
  <TCols>
    <TColHeader>GPG ID</TColHeader>
    <TColHeader>Last Fast Scan</TColHeader>
    <TColHeader>Last Full Scan</TColHeader>
    <TColHeader>Last Heartbeat</TColHeader>
    <TColHeader>Last Update</TColHeader>
    <TColHeader>Version</TColHeader>
  </TCols>
  <TRow>
    <TData>AB5D7197</TData>
    <TData>Up To Date</TData>
    <TData>Up To Date</TData>
    <TData>Up To Date</TData>
    <TData>21-JUL-2009</TData>
    <TData># release-2-15</TData>
  </TRow>
  <TRow>
    <TData>00862D98</TData>
    <TData>Up To Date</TData>
    <TData>Up To Date</TData>
    <TData>Up To Date</TData>
    <TData>01-JUL-2009</TData>
    <TData># test-0-118</TData>
  </TRow>
```

XML (actual)

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<XMLReport>
```

```
  <ReportHeader>
```

```
    <ReportTitle>XML Transformation Test</ReportTitle>
```

```
    <Author>Rich Casella</Author>
```

```
    <DeptCode>AO</DeptCode>
```

```
    <date>Oct 21 14:47:43 EDT 2009</date>
```

```
    <Source>Ordo Database</Source>
```

```
    <Summary>This is a test of HTML to PDF translation using XML as the mark up language and XSLT and the transformation mechanism.
```

All HTML code is transformed to a standard format, saved to a temporary file, and emailed to the chosen recipient. </Summary>

```
  </ReportHeader>
```

```
  <ReportTable>
```

```
    <TCaption>XML TEST</TCaption>
```

```
    <TNumCols>6</TNumCols>
```

```
    <Orientation>Portrait</Orientation>
```

```
    <TCols>
```

```
      <TColHeader>GPG ID</TColHeader>
```

```
      <TColHeader>Last Fast Scan</TColHeader>
```

```
      <TColHeader>Last Full Scan</TColHeader>
```

```
      <TColHeader>Last Heartbeat</TColHeader>
```

```
      <TColHeader>Last Update</TColHeader>
```

```
      <TColHeader>Version</TColHeader>
```

```
    </TCols>
```

```
    <TRow>
```

```
      <TData>B2F926DE</TData>
```

```
      <TData>04-OCT-2009</TData>
```

```
      <TData>19-OCT-2009</TData>
```

```
      <TData>20-OCT-2009</TData>
```

```
      <TData>Up To Date</TData>
```

```
      <TData>Up To Date</TData>
```

```
    </TRow>
```

```
    <TRow>
```

```
      <TData>53BDE375</TData>
```

```
      <TData>17-MAR-2009</TData>
```

```
      <TData>20-OCT-2009</TData>
```

```
      <TData>20-OCT-2009</TData>
```

```
      <TData>Up To Date</TData>
```

```
      <TData>Up To Date</TData>
```

```
    </TRow>
```

```
# Transform the content to XML
$xml::HTMLTable2XML(
  "XML Transformation Test",      # Report Title
  "Rich Casella",                # Author
  "AO",                           # Department Code
  "$date",                        # Date
  "Ordo Database",               # Content Source
  "$summary",                    # Report Summary
  "XML TEST",                    # Table Caption
  1,                              # 1=Portrait, 0=Landscape
  6,                              # Number of Columns
  $content,                      # Data for report
  \$xml                          # XML transformed from HTML
);
```

XSL Transformation (XSLT)

```
<xsl:template match="TRow">
  <xsl:variable name="rowNum">
    <xsl:element name="rowNum">
      <xsl:number count="TRow"/>
    </xsl:element>
  </xsl:variable>
  <xsl:if test="true() = ($rowNum mod 2)">
    <fo:table-row color="black">
      <fo:table-cell border="solid black 1px">
        <fo:block font-size="8pt" end-indent="0.05in" text-align="right">
          <xsl:value-of select="$rowNum"/>
        </fo:block>
      </fo:table-cell>
      <xsl:for-each select="TData">
        <fo:table-cell border="solid black 1px">
          <fo:block font-size="8pt" start-indent="0.05in">
            <xsl:value-of select="."/>
          </fo:block>
        </fo:table-cell>
      </xsl:for-each>
    </fo:table-row>
  </xsl:if>
  <xsl:if test="false() = ($rowNum mod 2)">
    <fo:table-row color="black" background-color="#B4C3DE">
```

PDF Report

Cover page (Portrait)



XML Transformation Test

Rich Casella

August 2, 1975

Department: AO
Source: Ordo Database

Summary

This is a test of HTML to PDF translation using XML as the mark up language and XSLT and the transformation mechanism. All HTML code is transformed to a standard format, saved to a temporary file, and emailed to the chosen recipient.

PDF Report

Table Data (Portrait)

XML TEST

Num	GPG ID	Last Fast Scan	Last Full Scan	Last Heartbeat	Last Update	Version
1	AB5D7197	Up To Date	Up To Date	Up To Date	21-JUL-2009	# release-2-15
2	00862D98	Up To Date	Up To Date	Up To Date	01-JUL-2009	# test-0-118
3	76711B9E	Up To Date	Up To Date	Up To Date	29-JUN-2009	# release-1-9
4	59B91A8D	Up To Date	Up To Date	Up To Date	17-JUL-2009	# release-2-15
5	226D60AC	Up To Date	Up To Date	Up To Date	30-JUN-2009	# release-2-8
6	1B987B75	Up To Date	Up To Date	Up To Date	16-JUL-2009	# release-1-55
7	FF5D647A	Up To Date	Up To Date	Up To Date	13-JUL-2009	# release-1-39
8	AB5D7197	Up To Date	Up To Date	Up To Date	21-JUL-2009	# release-2-15
9	00862D98	Up To Date	Up To Date	Up To Date	01-JUL-2009	# test-0-118

-
-
-

54	226D60AC	Up To Date	Up To Date	Up To Date	30-JUN-2009	# release-2-8
55	1B987B75	Up To Date	Up To Date	Up To Date	16-JUL-2009	# release-1-55
56	FF5D647A	Up To Date	Up To Date	Up To Date	13-JUL-2009	# release-1-39
57	AB5D7197	Up To Date	Up To Date	Up To Date	21-JUL-2009	# release-2-15
58	00862D98	Up To Date	Up To Date	Up To Date	01-JUL-2009	# test-0-118
59	76711B9E	Up To Date	Up To Date	Up To Date	29-JUN-2009	# release-1-9
60	59B91A8D	Up To Date	Up To Date	Up To Date	17-JUL-2009	# release-2-15
61	226D60AC	Up To Date	Up To Date	Up To Date	30-JUN-2009	# release-2-8
62	1B987B75	Up To Date	Up To Date	Up To Date	16-JUL-2009	# release-1-55
63	FF5D647A	Up To Date	Up To Date	Up To Date	13-JUL-2009	# release-1-39

Source: Ordo Database

Report Generated: August 2, 1975

Page 2

Example 2

Web page

UNIX Derivative OS Distribution Across BNL

This report shows the distribution of Unix derived Operating Systems at Brookhaven by department code. Blank table entries indicate there are no machines with that OS in that department. All columns are totaled and the totals are listed on the bottom line.

DEPT	TOTAL	AIX	Darwin	HP-UX	IRIX	IRIX64	Linux	NO_KERNEL	SunOS
AD	252	0	49	0	0	0	182	0	21
AM	8	0	4	0	0	0	4	0	0
AO	254	0	19	0	0	0	213	0	22
BD	1	0	0	0	0	0	1	0	0
BO	115	0	12	0	0	12	90	0	1
CC	22	0	3	0	1	0	18	0	0
CO	28	0	4	0	0	2	22	0	0
DA	2	0	2	0	0	0	0	0	0
DB	4	0	3	0	0	0	1	0	0
DE	8	0	0	0	0	0	8	0	0
DJ	1	0	1	0	0	0	0	0	0
DK	5	0	5	0	0	0	0	0	0
DL	14	0	3	0	0	0	11	0	0
EE	67	0	12	0	1	0	42	0	12

New XML Library Calls

Two Step Process

```
# Transform the content to XML, adding header information for report
```

```
&xml::HTMLTable2XML(  
    "XML Transformation Test",      # Report Title  
    "Rich Casella",                # Author  
    "AO",                           # Department Code  
    "$date",                        # Date  
    "Ordo Database",               # Content Source  
    "$summary",                    # Report Summary  
    "XML TEST",                     # Table Caption  
    1,                              # 1=Portrait, 0=Landscape  
    6,                              # Number of Columns  
    $content,                       # Data for report  
    \$xml                           # XML transformed from HTML  
);
```

```
# Transform the XML to PDF and email it to recipient(s)
```

```
&xml::XMLTable2PDF($xml, 'rac@bnl.gov');
```

PDF Report 2

Cover Page (Landscape)



UNIX Derivative OS Distribution Across BNL

Rich Casella

Wed Nov 4 12:27:13 EST 2009

Department: ALL Departments
Source: Ordo Database

Summary

This report shows the distribution of Unix derived Operating Systems at Brookhaven by department code. Blank table entries indicate there are no machines with that OS in that department. All columns are totaled and the totals are listed on the bottom line.

PDF Report 2

Table Data (Landscape)

OS Distributions

Num	DEPT	TOTAL	AIX	Darwin	HP-UX	IRIX	IRIX64	Linux	NO_KERNEL	SunOS
1	AD	252		49				182		21
2	AM	8		4				4		
3	AO	254		19				213		22
4	BD	1						1		
5	BO	115		12			12	90		1
6	CC	22		3		1		18		
7	CO	28		4			2	22		
8	DA	2		2						
9	DB	4		3				1		
10	DE	8						8		
11	DJ	1		1						
12	DK	5		5						
13	DL	14		3				11		
14	EE	67		12		1		42		12
15	ES	2		1				1		
16	EU	1		1						
17	ID	1						1		
18	IO	21		2				15		4
19	LP	1		1						
20	LS	149		14	40			94		1
21	LT	40		8				32		
22	MO	28		16				6		6
23	NC	29		15				14		
24	NE	22		6				16		
25	NN	11		3				8		
26	NO_DEPT	411	4	14				385	2	6
27	PA	16		15				1		
28	PG	18		18						
29	PM	32		8		1		23		
30	PO	3279	10	86		1	2	3057		123
31	QA	1						1		
32	RP	1						1		
33	TOTALS	4844	14	325	40	4	16	4247	2	196

Source: Orbis Database
Report Generated: Wed Nov 4 12:27:13 EST 2009

Page 2

Problems to Work Out

- Dynamic column sizing
 - Don't have to pass number of columns
 - Reasonable size automatically calculated (can be overridden)
- More complex content
 - Individual cell color (or other) attributes
 - Formats other than HTML TABLE
- Other formatting options
 - CSV Report
 - WML - Mobile? (Maybe not)
 - Direct to XML from DB, then to other formats, depending on target
 - XHTML
 - WML (mobile devices)
 - others (HTML5)?

Resources

- W3C XSL Documentation: <http://www.w3.org/TR/xsl/>
- Xalan: <http://xalan.apache.org/>
- FOP: <http://xmlgraphics.apache.org/fop/>
- XML Editor: <http://www.oxygenxml.com/>
- Contact info:
 - Rich Casella
 - Email: rac@bnl.gov
 - Twitter: @RACasella